

Using LAN card with E128 with no RAM card

LAN Hardware settings

1. JB1 - 1.2 Latch decoding
2. JB2 - 1.2 LAN decoding
3. JB3 – 1.2 Reset select
4. JB4 – 2.3 Input Power select
5. JB7 – 2.3 Latch decoding

Website links: <http://www.smsc.com/main/catalog/lan91c111.html>

The LAN card uses the SMCS LAN91C111/113 a 10/100 Non_PCi Ethernet Single Chip MAC + PHY interface. The LAN chip select is normally decoded by default at address \$300. At this location it is in conflict with the internal RAM resources of the E128 so it must be moved to address \$8000 that way it is accessible in expanded mode. This process would allow the LAN card to be access at \$8000 of banks \$00 - until flash is present at PPAGE \$38.

The E128 is 1st powered up in single chip mode making PORTA, PORTB and PORTE as normal I/O port pins. The address and DATA of the LAN are manipulated manually by port toggling the latches as to output the address and DATA. Port E port bit R/W*, LSTRB*, XCS*, ECLK are bit toggled to send address and DATA to the LAN chip. Below is an example code of how to do this. Once the LAN is moved to address \$8000, the MCU is then set to expanded wide mode to access it.

EtherMove.asm

```
;  
;Before entering expanded mode this routine uses the ports as GPIO  
;to command the Ethernet device to 0x8000.
```

EtherMove

```
    bset    PORTQ,%01000000    ;Reset the LAN card  
    bset    DDRQ,%01000000    ;Ethernet controller reset  
controlled via PTQ6  
    bclr    PORTQ,%01000000  
  
    jsr     delay100ns        ;100ns Reset delay  
  
    movb   #$1C,DDRE  
    movb   #$FF,DDRA  
    movb   #$FF,DDRB  
  
    bset    DDRK,%01000000    ;PortK bit 6  
    bset    PORTK,%01000000  
  
    bset    DDRK,%00010000    ;PortK bit 4  
    bset    PORTK,%00010000  
  
    bset    PEAR,NECLK        ;PortE bit NECLK  
  
    bset    PORTQ,%01000000    ;PortK bit 6 Ethernet  
controlled RESET OFF  
    jsr     delay50ms         ;50ms Reset delay
```

```

movw    #$030E,WriteAddress
movw    #$0100,WriteData
jsr     WriteLAN

movw    #$0302,WriteAddress
movw    #$0080,WriteData      ;Move LAN to $8000
jsr     WriteLAN

rts

```

WriteLAN

```

bset    PORTE,XCSHI+RWHI+LSTRBHI
bclr    PORTE,ECLKLO

ldd     WriteAddress
staa    PORTA
stan    PORTB

bclr    PORTE,XCSLO+RWLO+LSTRBLO
bset    PORTE,ECLKHI

ldd     WriteData
staa    PORTA
stab    PORTB

bset    PORTE,XCSHI+RWHI+LSTRBHI
bclr    PORTE,ECLKLO

rts

```

Using LAN card with E128 with 128K RAM card

The LAN card will only work with 256Kbyte RAM card. It will not work with 1Mbyte RAM card. There is a minor adjustment to the LAN board in order for RAM and LAN to work together. JB1 pin 2 must be connected with XADDR18.

LAN Hardware settings

1. JB1 - Jumper XADDR18 to pin 2
2. JB2 - 1.2 LAN decoding
3. JB3 – 1.2 Reset select
4. JB4 – 2.3 Input Power select
5. JB7 – 2.3 Latch decoding

Website links: <http://www.smsc.com/main/catalog/lan91c111.html>

RAM Hardware settings

1. JB1 – 1.2 CS1* RAM chip select
2. JB2 - 1.2 CS2 RAM chip select
3. JB6 – 1.2 A15/XA15 Address select
4. JB7 – 1.2 A15/XA15 Address select

The LAN card uses the SMCS LAN91C111/113 a 10/100 Non_PcI Ethernet Single Chip MAC + PHY interface. The LAN chip select is normally decoded by default at address \$300. At this location it is in conflict with the internal RAM resources of the E128 so it must be moved to address \$8000 that way it is accessible in expanded mode. This process would allow the LAN card to be access at \$8000 of banks \$00 - \$0F and the RAM is at \$8000 of banks \$10 until flash is present at PPAGE \$38.

The E128 is 1st powered up in single chip mode making PORTA, PORTB and PORTE as normal I/O port pins. The address and DATA of the LAN are manipulated manually by port toggling the latches as to output the address and DATA. Port E port bit R/W*, LSTRB*, XCS*, ECLK are bit toggled to send address and DATA to the LAN chip. Below is an example code of how to do this. Once the LAN is moved to address \$8000, the MCU is then set to expanded wide mode to access it.

EtherMove.asm

```
;
;Before entering expanded mode this routine uses the ports as GPIO
;to command the Ethernet device to 0x8000.
```

EtherMove

```
    bset    PORTQ,%01000000    ;Reset the LAN card
    bset    DDRQ,%01000000    ;Ethernet controller reset
controlled via PTQ6
    bclr    PORTQ,%01000000

    jsr     delay100ns        ;100ns Reset delay

    movb    #$1C,DDRE
    movb    #$FF,DDRA
    movb    #$FF,DDRB

    bset    DDRK,%01000000    ;PortK bit 6
    bset    PORTK,%01000000

    bset    DDRK,%00010000    ;PortK bit 4
    bset    PORTK,%00010000

    bset    PEAR,NECLK        ;PortE bit NECLK

    bset    PORTQ,%01000000    ;PortK bit 6 Ethernet
controller RESET OFF
    jsr     delay50ms         ;50ms Reset delay

    movw    #$030E,WriteAddress
    movw    #$0100,WriteData
    jsr     WriteLAN

    movw    #$0302,WriteAddress
    movw    #$0080,WriteData    ;Move LAN to $8000
    jsr     WriteLAN

    rts
```

WriteLAN

```
    bset    PORTE,XCSHI+RWHI+LSTRBHI
```

```
bclr    PORTE,ECLKLO

ldd     WriteAddress
staa    PORTA
stan    PORTB

bclr    PORTE,XCSLO+RWLO+LSTRBLO
bset    PORTE,ECLKHI

ldd     WriteData
staa    PORTA
stab    PORTB

bset    PORTE,XCSHI+RWHI+LSTRBHI
bclr    PORTE,ECLKLO

rts
```